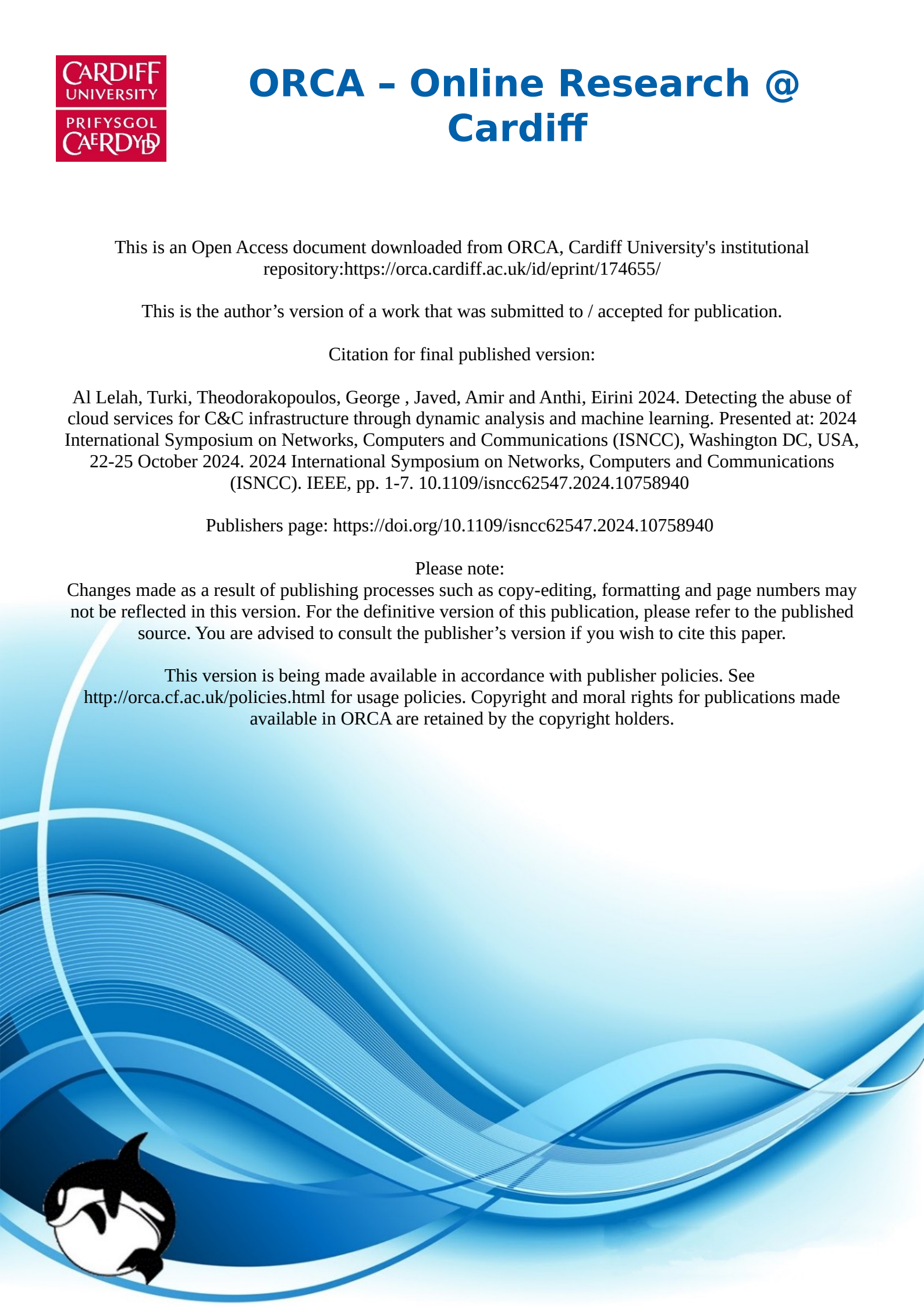

Detecting the Abuse of Cloud Services for C&C Infrastructure Through Dynamic Analysis and Machine Learning

Turki Al lelah, George Theodorakopoulos, Amir Javed, Eirini Anthi

School of Computer Science and Informatics
Cardiff University
Cardiff, UK
Email: {allelaht,theodorakopoulosg,javeda7,anthies}@cardiff.ac.uk

***Abstract*—Cybercriminals increasingly abuse cloud and legitimate services (CLS) as covert command and control (C&C) infrastructure to orchestrate malicious operations and evade detection. This paper addresses the critical challenge of detecting such abuse of cloud platforms. We introduce a detection system that integrates dynamic analysis with Machine Learning (ML) to accurately distinguish between benign and malicious interactions with cloud services. By utilising a comprehensive data set from VirusTotal, the system uses advanced feature extraction techniques from both host behaviour and network traffic, using Cuckoo and Triage sandboxes to extract behaviors, to develop a detection model. The results demonstrate that the model achieves nearly 98% accuracy in identifying cloud service abuse, substantially outperforming previous efforts. Furthermore, we evaluate the model's robustness against adversarial attacks that aim to decrease accuracy by manipulating the feature values. Comparative evaluations show that our method maintains a higher detection accuracy under attack compared to related systems.**

***Index Terms*—Cloud computing security, Command and control, Malware detection, Machine learning, Dynamic analysis, Adversarial machine learning attack**

I. Introduction

The digital landscape is undergoing a rapid transformation, highlighted by the increasing reliance on cloud and legitimate services (CLS) in critical sectors such as healthcare, finance, education and government. This dependence is driving the global public cloud services market towards an anticipated value of $2.5 trillion by 2031, according to research by Allied Market Research (AMR) [1], up from $551.8 billion in 2021. CLS deployment can save organisations more than 35% of their annual operating costs. By 2028, cloud computing will shift from being a technology disruptor to becoming a necessary component for maintaining business competitiveness. Gartner [2] predicts that more than 50% of enterprises will use industry cloud platforms by 2028 to accelerate their business initiatives. However, despite these benefits, cloud computing has also become a fertile ground for cyber threats, with malware authors leveraging cloud services to orchestrate malicious operations, such as abusing them as C&C infrastructure. In so doing, they exploit the inherent trust between CLS providers and their users to complicate detection significantly.

A study by Al lelah *et al.* [3] highlights the increasing abuse of cloud-based services, facilitated by various attack techniques. The study indicates a substantial increase in cloud-based abuses, yet the focus on developing detection strategies remains limited. Furthermore, reports [4, 5] confirm that threat actors continue to abuse CLS. In particular, state-sponsored actors increasingly to host C&C infrastructures for cyber-espionage campaigns. This trend underscores the urgent need for more advanced detection techniques. Abusing CLS for C&C allows attackers to:

- Conceal their operations: By blending malicious traffic with legitimate activity, cybercriminals make it harder for security systems to detect their presence.
- Launch widespread attacks: Cloud services offer them access to a large pool of potential victims, making it easier to launch large-scale attacks.
- Increase resilience: The distributed nature of cloud services makes it more difficult to take down their C&C infrastructure.

Therefore, this research addresses a critical challenge: the detection of malware that abuse the CLS as C&C infrastructure. This issue poses a substantial concern for cloud service providers and users alike. To address this challenge, we present a detection system that integrates dynamic analysis and ML. Our proposed model demonstrates nearly 98% detection accuracy, significantly surpassing existing related works. An essential aspect of our methodology is not merely focus on the detection, but also the thorough evaluation of our model's resilience against sophisticated adversarial attacks that manipulate the feature values to reduce the detection accuracy.

II. Exploitation of Cloud Services in Malware Attacks

Malware that abuses CLS turns them into a tool to manage compromised systems, distribute malware, or launch Denial

979-8-3503-6491-0/24/$31.00 ©2024 IEEE

of Service (DoS) attacks, among other malicious activities. Leveraging the trusted reputation and widespread reach of cloud services, attackers can obscure their activities, making it more challenging for traditional security measures to detect and mitigate such threats. Several examples illustrate this growing threat. vSingle [6] is a malware that traditionally appears as a PE file (.exe) that loads into the memory of a target and abuses GitHub repositories to perform its C&C operation. Similarly, Google Calendar RAT (GCR) [7, 8] is another malware that uses the Calendar service as a C&C infrastructure. This tool creates a covert channel by abusing event descriptions in Google Calendar, making it difficult for defenders to detect suspicious activity. The GCR, running on a compromised machine, periodically polls the calendar event description for new commands, executes those commands on the target device, and then updates the event description with command output. In addition, Google Drive was abused by malware called RogueRobin [9] that can establish an alternative C&C channel using the Google Drive API. Furthermore, Trendmicro's researcher [10] reported a Remote Access Trojan (RAT) that uses Google Drive for its malicious operations. The Advanced Persistent Threat (APT) group behind this malware is known as Pawn Storm. Moreover, WIP26 [11] is a threat cluster that is heavily based on the public cloud infrastructure. WIP26 uses two backdoors, CMD365 and CMDEmber, which abuse Microsoft 365 Mail and Google Firebase services for C&C purposes. Microsoft Azure and Dropbox instances are also used for data exfiltration and malware hosting. The consequences of these attacks can be profound, ranging from data breaches and loss of service integrity to significant financial and reputational damage. The process of abusing cloud services as C&C infrastructure is illustrated in Figure 1. In this workflow, the Botmaster, who controls the malicious activities, releases commands to the cloud service. These commands are then checked, retrieved, and executed by the Compromised Machine under the Botmaster's control. The results of these activities are collected using the Cloud Service as an intermediary platform.

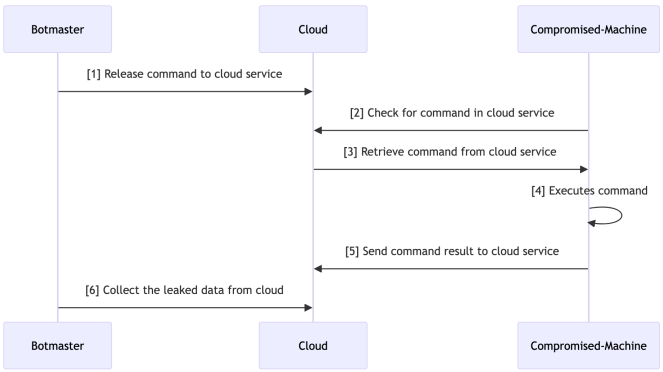

Fig. 1. Workflow of Cloud Services Abuse as C&C Infrastructure

III. Related Work

The related work section provides a concise overview of the current methods and challenges in CLS abuse detecting. It is divided into three main categories: ML-Based Detection, Rule-Based Detection, and Behavior Tree-Based Detection, highlighting both advantages and limitations.

A. ML-Based Detection

Al lelah *et al.* [12] applied ML to detect bots that use cloud services for C&C purposes, achieving a detection rate of 98. 26% with random forest classifiers. Their work involved creating a new dataset and extracting features from PE files. However, they noted challenges, such as vulnerability to the adversarial attack, which reduced accuracy to 84%, and issues with encrypted PE files hindering feature extraction. Ji *et al.* [13] analyzed abusive social bots, using spatial and temporal correlations to identify evasion patterns, focusing on botnets like Twitterbot (Singh [14]), Twebot (Burghouwt *et al.* [15]), Yazanbot (Boshmaf *et al.* [16]), Nazbot (Kartaltepe *et al.* [17]), wbbot (Ji *et al.* [18]), and fbbot. They proposed an 18-feature detection strategy to track patterns and behavior sequences, achieving insight into bot evasion tactics. However, the applicability of their study to other bots could be limited because of its focus on only six social networking bots. Ahmadi *et al.* [19] proposed a technique using Flowdroid to analyze Google Cloud Messaging (GCM) in Android apps for C&C detection. They trained an ML model with GCM-specific features, effectively distinguishing between benign and malicious apps. However, the technique faces potential evasion through obfuscation and is limited to Android, not applicable to other platforms like Windows.

B. Rule-Based Detection

Kartaltepe *et al.* [17] introduced a dual-level abuse detection system that focuses on botnet identification. This system utilizes client-side features such as self-concealment, anomalous network traffic, and suspicious origin sources. Additionally, it employs server-side analysis of text-encoded messages using the J48 decision tree algorithm. The primary aim of their approach is to distinguish bots from humans based on behavior and GUI interactions. They presum social platform connections and encoded messages as potentially suspicious indicators. However, the system has its limitations. Specifically, it lacks real-time detection capabilities and may be susceptible to evasion by sophisticated techniques like image steganography. Furthermore, Vo *et al.* [20] introduced API Verifier, a tool that leverages CAPTCHA verification to authenticate access to social media accounts using MAC addresses. This tool is designed to discern whether an API call originates from a human or a bot, providing a protective layer against automated bot activities. However, it's susceptible to relay attacks and MAC address spoofing. Ghanadi *et al.* [21] conducted an in-depth study of stego-botnets that employ steganographic images on social networks for C&C operations. They introduced a system called SocialClymene, which is designed to detect hidden botnets in social networks that use stego-images. The objective is to identify botnets by examining the users' behavior and their past involvement in suspicious activities. However, these detection methodologies do have certain limitations: (i)

the system may fail to detect new botnets that do not have a history of suspicious behavior, and (ii) accurately determining a user's reputation can be challenging, particularly in dynamic online environments where reputations can change rapidly.

C. Behavior Tree-Based Detection

Yuede *et al.* [22] proposed a behavior tree-based detection framework to identify social bots by monitoring host activities, consisting of behavior monitoring, analysis, and detection components. They created a botnet, wbbot, for analysis, using real-world and researcher-collected social bot samples to build a template library for comparison. However, the framework has limitations, including a high false positive rate of 29.6% and potential evasion by attackers using multi-process strategies or spreading malicious behaviors over time. Burghouwt *et al.* [23] introduced a causality detection mechanism to identify Twitter-based C&C communication by correlating user activity with network traffic, deeming non-human-driven traffic as suspicious. This method uses a temporal window to distinguish between legitimate user actions and bot-initiated activities. However, it faces challenges such as misidentifying legitimate API calls as suspicious, inaccuracies in timing measurements due to hardware and software variances, and the potential for advanced bots to evade detection by mimicking user behaviors.

In conclusion, these works collectively concentrate on the application of rule-based and behaviour tree-based detection, as well as static malware analysis combined with ML techniques, within the context of social networking platforms and cloud platforms. However, dynamic malware analysis techniques for identifying abuse in cloud environments are often overlooked, leading to a gap in the detection of CLS abuse as C&C infrastructure. To address this, we propose a detection technique that leverages dynamic analysis and ML, specifically designed to identify such abuse.

IV. Methodology

The methodology section is divided into two main components. The first, "Data Collection," details the process of assembling the dataset. The second part introduces a groundbreaking technique called "Dual-Sandboxing for Advanced Feature Extraction," which employs the Cuckoo and Triage tools to enhance feature extraction. This dual-sandboxing approach gathers a broad spectrum of features related to both system interactions and network activities.

A. Data Collection

In this study, we sourced a dataset from VirusTotal spanning 2017 to 2021, with a specific focus on Portable Executable (PE) files that misused the CLS for C&C activities. In particular, we extracted malware samples demonstrating CLS domain interactions using the VirusTotal Intelligence Agent [24] and a Python script. Any samples not connecting to CLS domains were excluded. Additionally, benign samples were obtained from CNET [25] and SourceForge [26] and were thoroughly verified to be safe and exhibit internet activity through rigorous testing in VirusTotal and Cuckoo sandboxes. Only benign samples with no antivirus detections and confirmed internet connectivity were carefully included in the final data set. Consequently, this meticulous approach yielded a comprehensive dataset of 2508 malicious and 2363 benign samples, aimed at maintaining balance to avoid any classifier bias.

B. Dual-Sandboxing for Advanced Feature Extraction

In our dataset, our objective is to extract features for ML applications. To this end, we introduce an innovative Dual-Sandboxing (DS) technique that combines the strengths of sandbox environments: Cuckoo and Triage. This method is tailored to capture a broad spectrum of features:

1) Host-based Features: These originate from the malware operations, such as Application Programming Interface (API) calls between the sample and the host Operating System (OS), file activities, and interactions with the system. We employ Cuckoo[27] to obtain a comprehensive footprint of the malware's behaviour within the host environment.

2) Network-based Features: These relate to the malware's interactions with external entities, including cloud services and the Internet. For this aspect, we use Triage[28], which operates within a cloud-based infrastructure and is equipped with sophisticated antievasion techniques, to observe and analyse network behaviours between the host and the Cloud Layer Services (CLS), pinpointing potential abuse attempts. Figure 2 illustrates the operational framework of the DS technique, highlighting the synergy between Cuckoo and Triage sandboxes in capturing diverse malware features.

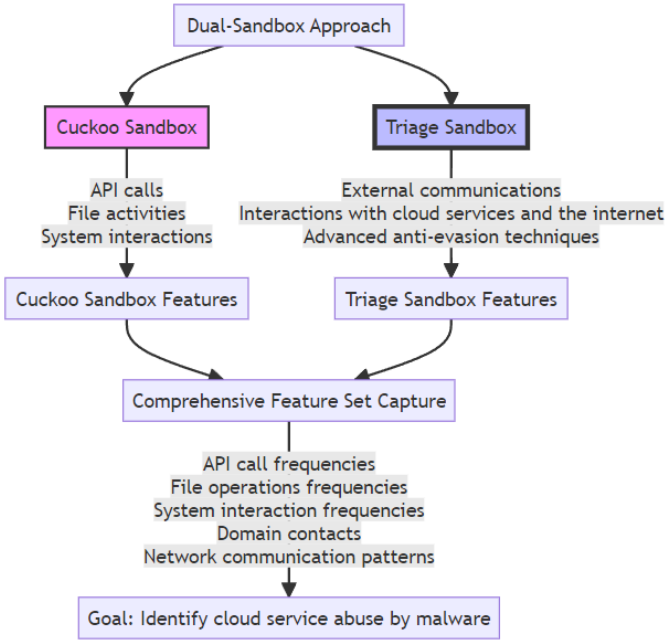

Fig. 2. Illustration of Dual-Approach Sandboxing Technique for Enhanced Feature Extraction

By integrating these insights from both the Cuckoo and Triage sandboxes, we generate a detailed dataset that encompasses both internal operations and external communications.

This dataset includes information on API call frequencies, file operation frequencies, system interaction patterns, domain communication logs, and network traffic flows. This comprehensive feature set is essential to identify CLS abuse by malware, marking a significant advancement in efforts to protect cloud-based services.

C. Feature Selection

To evaluate whether all set of features should be included or only sub set, we employed both filter-based (Information Gain and ReliefF) and wrapper-based (Random Forest and J48) feature selection strategies. Despite these efforts, these methods were unable to surpass the 98% accuracy rate achieved with the comprehensive suite of 99 features derived from our Dual-Sandbox approach. This outcome highlights the crucial role and collective importance of each feature in accurately differentiating between malicious and benign samples. Considering the complexity and moderate size of our set of features, our analysis suggests that a reduction in features is not essential to achieve optimal detection performance.

V. Evaluation

A. Detection Accuracy

We conducted a meticulous evaluation of various algorithms, including Decision Tree (J48), Naïve Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest. In our approach, we applied both a 10-fold cross-validation and a 70/30 split to our dataset. This comprehensive examination, spanning multiple validation techniques, ensured the robustness of our predictive models. The result is shown in Table I. In comparing the two validation techniques, the Random Forest algorithm consistently outperformed other models, achieving the highest accuracy of 97.95% (70/30 split) and 97.31% (10-fold cross-validation). This performance suggests that Random Forest effectively captures complex relationships within the dataset, thereby demonstrating its suitability for the task of detecting cloud service abuse by malware. On the other hand, the J48 decision tree model also performed well, achieving an accuracy of 96.51% (70/30 split) and 95.96% (10-fold cross-validation). Although slightly lower than Random Forest, J48 still showed robust performance.

TABLE I
Detection Accuracy of ML Algorithms

Algorithm	70/30 Split (%)	10-fold CV (%)
Random Forest	**97.95**	97.31
J48	96.51	95.96
NB	71.32	71.81
K-NN	96.10	95.98
SVM	93.50	92.98

B. Comparison to Related Work

While there is an abundance of literature on malware detection systems that utilise dynamic analysis, there appears to be a gap in research specifically aimed at detecting the abuse of cloud services as C&C infrastructure. Therefore, we compare our work with general dynamic analysis-based malware detection systems, Sethi et al.[29] and Shijo et al.[30]. The comparison serves a dual purpose: it not only emphasizes the enhancements and contributions our work introduces to the field through advanced feature engineering but also measures our system's performance against the established benchmarks of dynamic malware analysis.

1) Detection Accuracy Comparison: This comparative analysis, elaborated in Table II, utilises a diverse array of classifiers, including J48, RF, NB, KNN, and SVM. To ensure the reliability and generalisability of our findings, these classifiers were evaluated using two distinct validation methods: a 70:30 split and 10-fold cross-validation.

TABLE II
Comparison of Abuse Detection Accuracy with Related Works

Reference	J48		RF		NB		KNN		SVM	
	70:30	10-fold	70:30	10-fold	70:30	10-fold	70:30	10-fold	70:30	10-fold
Sethi et al.[29]	74.89	75.94	76.33	76.44	68.39	67.63	76.12	76.15	73.87	74.55
Shijo et al.[30]	71.90	72.04	72.70	73.08	67.90	69.80	71.83	72.70	71.30	73.26
Proposed work	96.51	95.96	**97.95**	97.31	71.32	71.81	96.10	95.98	93.50	92.98

Our findings reveal a significant improvement in detection accuracy with the proposed set of features. For instance, using the RF classifier, our system achieves an accuracy of 97.94% with a 70:30 split and 97.31% with 10-fold cross-validation. This is substantially higher compared to the accuracy reported in the works of Sethi et al. [29] and Shijo et al. [30], which underscores the advanced feature engineering using the DS. Furthermore, the results from the other classifiers corroborate the superiority of our proposed system. With the J48 classifier, our system consistently outperforms the related works with a 96.50% accuracy on a 70:30 split and a 95.95% with 10-fold cross-validation. Similar trends are observed with KNN and SVM classifiers, in which the proposed system exhibits robust performance.

2) Robustness Comparison: An essential aspect of our methodology involves evaluating our model's robustness against sophisticated adversarial attacks. In an adversarial attack, the attacker modifies feature values in the malware samples to decrease the model accuracy by making the malware samples resemble benign software. Table III presents the top 10 features most impacted by such attacks, demonstrating a significant decrease in detection accuracy. The feature num_ips caused the largest accuracy drop of 7.94% when its value for the malware was changed from the original malware value to 2, the corresponding benign value, thereby increasing the False Negative (FN) rate. Similarly, other features like total_dns_requests and total_udp_connections also led to noticeable accuracy declines of approximately 1-3% when altered. Across the top 10 features, the decrease in accuracy reveals vulnerabilities that could potentially be exploited to evade detection. However, even after the adversarial attack, the accuracy only decreased to 90% for these features, indicating some resilience of the model. The table provides a comparison of the original and

manipulated confusion matrices, delineating the precise impact on True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). In particular, a significant increase in FN suggests the potential for malware to be classified as benign, posing a threat to the cloud security infrastructure. Furthermore, the results of adversarial attacks on the detection accuracy of our proposed model, as well as related works by Sethi et al. [29] and Shijo et al. [30], are presented in Tables III, IV, and V respectively. The discussion focuses on the changes in the number of false negatives (FN) in the confusion matrix (CM), which reflect the model's ability to correctly identify malicious samples under FPT attack. For our proposed work, the feature 'num_ips' experienced the most significant drop in detection accuracy after feature perturbation testing (FPT), from 97.95% to 90.01%. This is indicative of the feature's importance in the detection algorithm and its vulnerability to adversarial manipulation. Other features also exhibited declines in detection accuracy, but to a lesser extent, suggesting a varied impact on the model's robustness. On the contrary, the model by Sethi et al. showed a uniform decrease in detection accuracy to 50.00% for all features when manipulated. The sharp increase in FN, especially for the feature 'f_LdrGetDllHandle'[1], raises concerns about the model's susceptibility to evasion techniques. Similarly, Shijo et al.'s model showed a similar trend, with the detection accuracy dropping to approximately 47.57% across all features. The increase in FN for features such as 'LdrGetDllHandle_NtTerminateProcess_NtTerminateProcess'[2] was substantial, suggesting that the model might not be robust against certain adversarial strategies.

TABLE III
TOP 10 IMPACTED FEATURES OF ADVERSARIAL ATTACK ON DETECTION ACCURACY, WITH A FOCUS ON CHANGES IN FN VALUES IN THE CONFUSION MATRIX (CM): PROPOSED WORK

Features	Minimizing Value	Original Accuracy	Post-FPT Accuracy	Original CM		Post-FPT CM	
num_ip	2	0.9795	90.01	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:403**	FP:327 TN:378
total_dns_requests	43	0.9795	0.9521	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:59**	FP:11 TN:722
num_domains	15	0.9795	0.9603	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:47**	FP:11 TN:734
total_file_failed	11	0.9795	0.9617	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:45**	FP:11 TN:736
total_tcp_connections	3	0.9795	0.9624	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:44**	FP:11 TN:737
total_directory_enumerated	3	0.9795	0.9631	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:43**	FP:11 TN:738
total_regkey_opened	3	0.9795	0.9637	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:42**	FP:11 TN:739
total_regkey_read	3	0.9795	0.9637	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:42**	FP:11 TN:739
total_http_requests	3	0.9795	0.9637	TP:688 FN:20	FP:10 TN:743	TP:670 **FN:42**	FP:11 TN:739

To this end, the proposed work suggests that the detection model experiences a decrease in accuracy under adversarial conditions, yet it still outperforms models from related works in terms of robustness. The resilience of our model, as observed by a minor drop in detection accuracy and fewer false negatives, underscores its efficacy in maintaining a higher detection rate. This comparative advantage is crucial, as it implies a more reliable performance in real-world scenarios where adversaries continuously evolve their tactics.

[1]Features represent the frequency of successful API (s_api), failed API (f_api) calls, and return codes from API (rc_api_return_code_value) for each sample.

[2]Features represent the frequency of 3-gram and 4-gram patterns in API calls within each sample.

TABLE IV
TOP 10 IMPACTED FEATURES OF ADVERSARIAL ATTACK ON DETECTION ACCURACY, WITH A FOCUS ON CHANGES IN FN VALUES IN THE CONFUSION MATRIX (CM): SETHI ET AL.[29]

Features	Minimizing Value	Original Accuracy	Post-FPT Accuracy	Original CM		Post-FPT CM	
f_LdrGetDllHandle	24	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
f_SetUnhandledExceptionFilter	1	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
rc_GetSystemMetrics_16	37	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
f_GetFileType	4	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
rc_LdrGetDllHandle_3221225781	24	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
f_RegOpenKeyExW	32	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
s_RegEnumKeyExW	12	0.7695	0.5000	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:404**	FP:327 TN:377
f_RegEnumKeyExW	5	0.7695	0.5007	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:403**	FP:327 TN:378
s_NtMapViewOfSection	1	0.7695	0.5007	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:403**	FP:327 TN:378
rc_NtOpenKey_0	3	0.7695	0.5007	TP:354 FN:10	FP:327 TN:771	TP:354 **FN:403**	FP:327 TN:378

TABLE V
TOP 10 IMPACTED FEATURES OF ADVERSARIAL ATTACK ON DETECTION ACCURACY, WITH A FOCUS ON CHANGES IN FN VALUES IN THE CONFUSION MATRIX (CM): SHIJO ET AL.[30]

Features	Minimizing Value	Original Accuracy	Post-FPT Accuracy	Original CM		Post-FPT CM	
LdrGetDllHandle_NtTerminateProcess _NtTerminateProcess	1	0.7332	0.4757	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:389**	FP:399 TN:375
LdrGetDllHandle_LdrGetDllHandle _NtTerminateProcess	1	0.7332	0.4757	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:389**	FP:399 TN:375
LdrGetDllHandle_LdrGetDllHandle _NtTerminateProcess_NtTerminateProcess	1	0.7332	0.4757	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:389**	FP:399 TN:375
NtAllocateVirtualMemory_NtAllocateVirtual Memory _NtAllocateVirtualMemory	69	0.7332	0.4770	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:387**	FP:399 TN:377
GetFileType_GetFileType _GetFileType	1	0.7332	0.4770	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:387**	FP:399 TN:377
NtClose_NtOpenKey _RegOpenKeyExW	5	0.7332	0.4770	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:387**	FP:399 TN:377
LdrGetDllHandle_LdrGetProcedureAddress _LdrGetDllHandle_LdrGetProcedureAddress	62	0.7332	0.4770	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:387**	FP:399 TN:377
NtAllocateVirtualMemory_NtAllocateVirtual Memory _NtAllocateVirtualMemory_Nt AllocateVirtualMemory	41	0.7332	0.4770	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:387**	FP:399 TN:377
NtClose_LdrLoadDll _LdrGetProcedureAddress	1	0.7332	0.4777	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:386**	FP:399 TN:378
SetUnhandledExceptionFilter_LdrGetDllHandle _LdrGetDllHandle	1	0.7332	0.4777	TP:340 FN:2	FP:399 TN:762	TP:340 **FN:386**	FP:399 TN:378

VI. CONCLUSIONS

This study addresses the significant challenge of cloud services being abused as C&C infrastructure by cybercriminals. It not only elucidates on the sophisticated tactics utilised by cybercriminals to abuse cloud platforms, but also introduces an innovative detection methodology that integrates dynamic analysis with ML algorithms to identify such abuses. Notably, by utilizing a comprehensive dataset from VirusTotal and advanced feature extraction methods from both system and network behaviors, our approach achieves a remarkable accuracy rate of approximately 98% in detecting such abuse, significantly outperforming existing work. Moreover, an evaluation of the model's resilience against adversarial attacks revealed a degree of vulnerability, with certain features exhibiting noticeable decreases in detection accuracy when manipulated, although the overall accuracy remained relatively high compared to existing works.

REFERENCES

[1] "Cloud services market size, share - 2031," https://www.alliedmarketresearch.com/cloud-services-market, March 2023, (Accessed on 08/31/2023).

[2] Gartner says cloud will become a business necessity by 2028. https://www.gartner.com/en/newsroom/press-releases/2023-11-29-gartner-says-cloud-will-become-a-business-necessity-by-2028. (Accessed on 12/31/2023).

[3] T. Al lelah, G. Theodorakopoulos, P. Reinecke, A. Javed, and E. Anthi, "Abuse of cloud-based and public legitimate services as command-and-control (c&c) infrastructure: a systematic literature review," *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 558–590, 2023.

[4] "Cloud service provider abuse explained – crowdstrike," https://www.crowdstrike.com/cybersecurity-101/cloud-security/cloud-service-provider-abuse/, (Accessed on 02/28/2024).

[5] "Cloud services are increasingly exploited for command and control in cyber espionage operations - infosecurity magazine," https://www.infosecurity-magazine.com/blogs/cloud-command-control-cyber/, (Accessed on 02/28/2024).

[6] vsingle is abusing github to communicate with the c2 server — infosec. https://resources.infosecinstitute.com/topics/vulnerabilities/vsingle-is-abusing-github-to-communicate-with-the-c2-server/. (Accessed on 08/31/2023).

[7] Hackers could abuse google calendar as a covert c2 channel — cyware alerts - hacker news. https://cyware.com/news/hackers-could-abuse-google-calendar-as-a-covert-c2-channel-8587a621/. (Accessed on 12/31/2023).

[8] Google warns how hackers could abuse calendar service as a covert c2 channel — cyber affairs. https://cyberaffairs.com/news/google-warns-how-hackers-could-abuse-calendar-service-as-a-covert-c2-channel/. (Accessed on 12/31/2023).

[9] Roguerobin malware uses google drive as c2 channel — threatpost. https://threatpost.com/roguerobin-google-drive-c2/141079/. (Accessed on 08/31/2023).

[10] Pawn storm's lack of sophistication as a strategy. https://www.trendmicro.com/en_us/research/20/l/pawn-storm-lack-of-sophistication-as-a-strategy.html. (Accessed on 08/31/2023).

[11] Wip26 espionage — threat actors abuse cloud infrastructure in targeted telco attacks - sentinelone. https://www.sentinelone.com/labs/wip26-espionage-threat-actors-abuse-cloud-infrastructure-in-targeted-telco-attacks/. (Accessed on 08/31/2023).

[12] T. Al lelah, G. Theodorakopoulos, A. Javed, and E. Anthi, "Machine learning detection of cloud services abuse as c&c infrastructure," *Journal of Cybersecurity and Privacy*, vol. 3, no. 4, pp. 858–881, 2023.

[13] Y. Ji, Y. He, X. Jiang, J. Cao, and Q. Li, "Combating the evasion mechanisms of social bots," *computers & security*, vol. 58, pp. 230–249, 2016.

[14] A. Singh, "Social networking for botnet command and control," 2012.

[15] P. Burghouwt, M. Spruit, and H. Sips, "Detection of covert botnet command and control channels by causal analysis of traffic flows," in *International Symposium on Cyberspace Safety and Security*. Springer, 2013, pp. 117–131.

[16] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Design and analysis of a social botnet," *Computer Networks*, vol. 57, no. 2, pp. 556–578, 2013.

[17] E. J. Kartaltepe, J. A. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: emerging threats and countermeasures," in *ACNS*. Springer, 2010, pp. 511–528.

[18] Y. Ji, Y. He, D. Zhu, Q. Li, and D. Guo, "A mulitiprocess mechanism of evading behavior-based bot detection approaches," in *International conference on information security practice and experience*. Springer, 2014, pp. 75–89.

[19] M. Ahmadi, B. Biggio, S. Arzt, D. Ariu, and G. Giacinto, "Detecting misuse of google cloud messaging in android badware," in *SPSM*, 2016, pp. 103–112.

[20] N. H. Vo and J. Pieprzyk, "Protecting web 2.0 services from botnet exploitations," in *2010 Second Cybercrime and Trustworthy Computing Workshop*. IEEE, 2010, pp. 18–28.

[21] M. Ghanadi and M. Abadi, "Socialclymene: A negative reputation system for covert botnet detection in social networks," in *7'th International Symposium on Telecommunications (IST'2014)*. IEEE, 2014, pp. 954–960.

[22] Y. Ji, Y. He, X. Jiang, and Q. Li, "Towards social botnet behavior detecting in the end host," in *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2014, pp. 320–327.

[23] P. Burghouwt, M. Spruit, and H. Sips, "Towards detection of botnet communication through social media by monitoring user activity," in *International Conference on Information Systems Security*. Springer, 2011, pp. 131–143.

[24] "Virustotal - intelligence overview," https://www.virustotal.com/gui/intelligence-overview, 2022, (Accessed on 02/25/2022).

[25] "Free software downloads and reviews for windows, android, mac, and ios – cnet download," 1996. [Online]. Available: https://download.cnet.com/

[26] "SourceForge.Net," https://sourceforge.net/projects/sourceforge/, 1999.

[27] "Cuckoo sandbox - automated malware analysis," https://cuckoosandbox.org/, February 2011, (Accessed on 05/18/2023).

[28] "Login — triage," https://tria.ge/, August 2018, (Accessed on 02/25/2022).

[29] K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," IEEE, pp. 1–4, 2019.

[30] P. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," *Procedia Computer Science*, vol. 46, pp. 804–811, 2015.